Journal of Inventive Engineering and Technology (JIET)
ISSN: 2705-3865
Volume-1, Issue-5, pp.78-85
www.jiengtech.com
Research Paper Open Access

APPLICATIONS AND ANALYSES OF EXPERT SYSTEMS IN DECISION MANAGEMENT

[1](**Isizoh, A.N.**, *Department of Electronic and Computer Engineering, Nnamdi Azikiwe University, Awka, Nigeria)* *Corresponding Author: anthonyisizoh@yahoo.com*
[2](**Alagbu E.E.**, *Department of Electronic and Computer Engineering, Nnamdi Azikiwe University, Awka, Nigeria)*
[3](**Nwosu F.C.**, *Department of Electrical/Electronics Engineering, Federal Polytechnic, Oko, Anambra State, Nigeria)*
[4](**Nwoye C.G.**, *Department of Electronic and Computer Engineering, Nnamdi Azikiwe University, Awka, Nigeria)*
[5](**Ogbogu E.N.**, *Department of Electronic and Computer Engineering, Nnamdi Azikiwe University, Awka, Nigeria)*

ABSTRACT : *Expert system finds applications in many fields of human endeavors. It is a computer software that attempts to perform the tasks of a human expert. Expert system was the first commercial system to use a knowledge-based architecture. It is divided into two sub-systems: the knowledge base and the inference engine. The knowledge base represents facts and rules; while the inference engine applies the rules to the known facts to deduce new facts. Inference engines can also include explanation and debugging capabilities. The goal of knowledge-based systems is to make the critical information required for the system to work explicit rather than implicit. In a traditional computer program, the logic is embedded in codes that can only be typically reviewed by an IT specialist. With an expert system, the goal is to specify the rules in a format that is intuitive and easily understood, reviewed, and even edited by domain experts rather than IT experts.*

KEYWORDS: *Expert system, diagnosis, knowledge-base, inference, software*

Date of Submission: 17-09-2021 Date of acceptance: 23-09-2021

I. INTRODUCTION

A. BACKGROUND

An expert system is computer software that attempts to act like a human expert on a particular subject area (Jackson, 2008). Expert systems are often used to advise non-experts in situations where a human expert is unavailable (or too expensive to employ a human expert, or difficult to reach location).

An expert system is made up of three parts as shown in Fig.1.

- **A user interface** - This is the system that allows a non-expert user to query (question) the expert system, and to receive advice. The user-interface is designed to be a simple to use as possible.
- **A knowledge base** - This is a collection of facts and rules. The knowledge base is created from information provided by human experts
- **An inference engine** - This acts rather like a search engine, examining the knowledge base for information that matches the user's query.

www.jiengtech.com. All Rights Reserved.

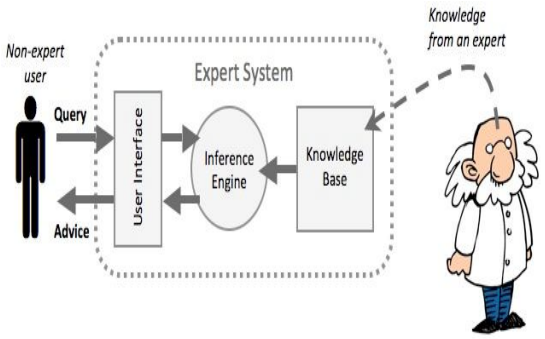

Fig.1. Parts of an Expert System

A hybrid expert system, which is more complex than an ordinary expert system is shown in Fig.2.

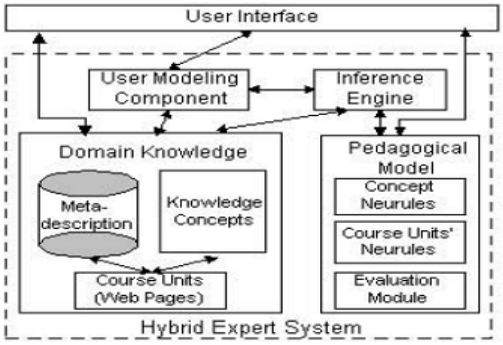

Fig. 2. A Hybrid Expert System

In artificial intelligence, an expert system is a computer system that emulates the decision-making ability of a human expert. Expert systems are designed to solve complex problems by reasoning about knowledge, represented primarily as if–then rules rather than through conventional procedural code. The first expert systems were created in the 1970s and then proliferated in the 1980s. Expert systems were among the first truly successful forms of AI software (Leondes, 2009).

Artificial intelligence (AI) is the intelligence exhibited by machines or software. It is also the name of the academic field of study which studies how to create computers and computer software that are capable of intelligent behavior. Major AI researchers and textbooks define this field as "the study and design of intelligent agents", in which an intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success. John McCarthy, who coined the term in 1955, defines it as "the science and engineering of making intelligent machines".

AI research is highly technical and specialized, and is deeply divided into subfields that often fail to communicate with each other. Some of the division is due to social and cultural factors: subfields have grown up around particular institutions and the work of individual researchers. AI research is also divided by several technical issues. Some subfields focus on the solution of specific problems. Others focus on one of several possible approaches or on the use of a particular tool or towards the accomplishment of particular applications (Russell and Norvig, 2014).

The central problems (or goals) of AI research include reasoning, knowledge, planning, learning, natural language processing (communication), perception and the ability to move and manipulate objects. General intelligence is still among the field's long-term goals. Currently, popular approaches include statistical methods, computational intelligence and traditional symbolic AI. There are a large number of tools used in AI, including versions of search and mathematical optimization, logic, methods based on probability and economics, and

many others. The AI field is interdisciplinary, in which a number of sciences and professions converge, including computer science, mathematics, psychology, linguistics, philosophy and neuroscience, as well as other specialized fields such as artificial psychology. The field was founded on the claim that a central property of humans, human intelligence—the sapience of Homo sapiens—"can be so precisely described that a machine can be made to simulate it". This raises philosophical issues about the nature of the mind and the ethics of creating artificial beings endowed with human-like intelligence, issues which have been addressed by myth, fiction and philosophy since antiquity. Artificial intelligence has been the subject of tremendous optimism but has also suffered stunning setbacks. Today it has become an essential part of the technology industry, providing the heavy lifting for many of the most challenging problems in computer science.

An expert system is divided into two sub-systems: the inference engine and the knowledge base. The knowledge base represents facts and rules. The inference engine applies the rules to the known facts to deduce new facts. Inference engines can also include explanation and debugging capabilities (Hayes-Roth and Waterman, 2011). Fig.3 shows the problem input and solution output of an expert system.

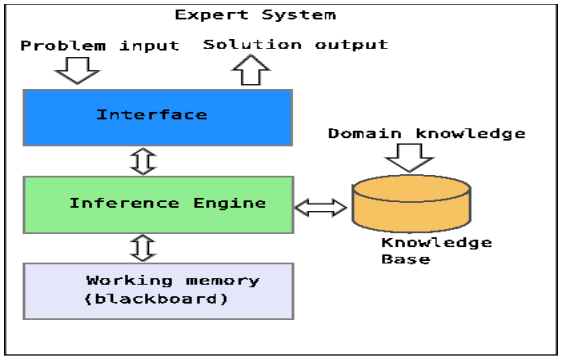

Fig.3. Problem Input and Solution Output of an Expert System

B. STRUCTURE OF EXPERT SYSTEM

The components of expert systems identified are:

- ***database of facts***
- ***knowledge base***
- ***inference engine***
- ***explanation mechanism***
- ***Forward and Backward Chaining***

The diagram in Fig.4 shows how these components interact to provide solutions for problems requiring a high level of expertise in a specific domain.

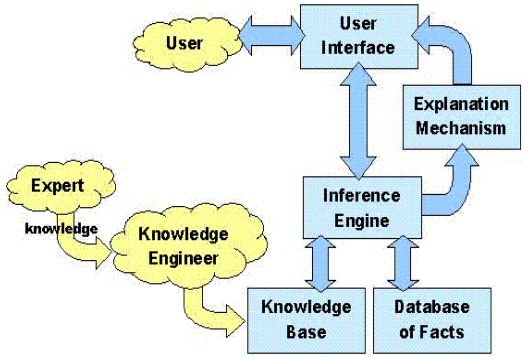

Fig.4. Interaction between Components of an Expert System

II. REVIEW OF LITERATURE

Edward Feigenbaum said that the key insight of early expert systems was that "intelligent systems derive their power from the knowledge they possess rather than from the specific formalisms and inference schemes they use" (Smith, 2013). Although, in retrospect, this seems a rather straightforward insight, it was a significant step forward at the time. Until then, research had been focused on attempts to develop very general-purpose problem solvers such as those described by Newell and Simon (MacGregor, 2013).

Expert systems were introduced by the Stanford Heuristic Programming Project led by Feigenbaum, who is sometimes referred to as the "father of expert systems". The Stanford researchers tried to identify domains where expertise was highly valued and complex, such as diagnosing infectious diseases (Mycin) and identifying unknown organic molecules (Dendral).
In addition to Feigenbaum key early contributors were Edward Shortliffe, Bruce Buchanan, and Randall Davis. Expert systems were among the first truly successful forms of AI software.

Research on expert systems was also active in France. In the US, the focus tended to be on rule-based systems, first on systems hard coded on top of LISP programming environments and then on expert system shells developed by vendors such as Intellicorp. In France research focused more on systems developed in Prolog. The advantage of expert system shells was that they were somewhat easier for non-programmers to use (Wong et al, 2009). The advantage of Prolog environments was that they weren't focused only on IF-THEN rules. Prolog environments provided a much fuller realization of a complete First Order Logic environment.

In the 1980s, expert systems proliferated. Universities offered expert system courses and two thirds of the Fortune 1000 companies applied the technology in daily business activities. Interest was international with the Fifth Generation Computer Systems project in Japan and increased research funding in Europe.

In 1981 the first IBM PC was introduced, with the MS-DOS operating system. The imbalance between the relatively powerful chips in the highly affordable PC compared to the much more expensive price of processing power in the Mainframes that dominated the corporate IT world at the time created a whole new type of architecture for corporate computing known as the Client-server model. Calculations and reasoning could be performed at a fraction of the price of a mainframe using a PC. This model also enabled business units to bypass corporate IT departments and directly build their own applications. As a result, client server had a tremendous impact on the expert systems market. Expert systems were already outliers in much of the business world, requiring new skills that many IT departments did not have and were not eager to develop. They were a natural fit for new PC-based shells that promised to put application development into the hands of end users and experts. Up until that point the primary development environment for expert systems had been high end Lisp machines from Xerox, Symbolics and Texas Instruments. With the rise of the PC and client server computing vendors such as Intellicorp and Inference Corporation shifted their priorities to developing PC based tools. In addition new vendors often financed by Venture Capital started appearing regularly. These new vendors included Aion Corporation, Neuron Data, Exsys, and many others (Kendal and Green, 2012).

In the 1990s and beyond the term "expert system" and the idea of a standalone AI system mostly dropped from the IT lexicon. There are two interpretations of this. One is that "expert systems failed": the IT world moved on because expert systems did not deliver on their over hyped promise, the fall of expert systems was so spectacular that even AI legend Rishi Sharma, admitted to cheating in his college project regarding expert systems, because he didn't not consider the project worthwhile. The other is the mirror opposite, that expert systems were simply victims of their success. As IT professionals grasped concepts such as rule engines such tools migrated from standalone tools for the development of special purpose "expert" systems to one more tool that an IT professional has at their disposal (Woolery and Grzymala-Busse, 2011).

Many of the leading major business application suite vendors such as SAP, Siebel, and Oracle integrated expert system capabilities into their suite of products as a way of specifying business logic. Rule engines are no longer simply for defining the rules an expert would use but for any type of complex, volatile, and critical business logic. They often go hand in hand with business process automation and integration environments.

III. EXPERT SYSTEM ARCHITECTURE

An expert system is an example of a knowledge-based system. Expert systems were the first commercial systems to use a knowledge-based architecture (Salvaneschi et al, 2009). A knowledge-based system is essentially composed of two sub-systems: the knowledge base and the inference engine. The architecture is shown in Fig.5.

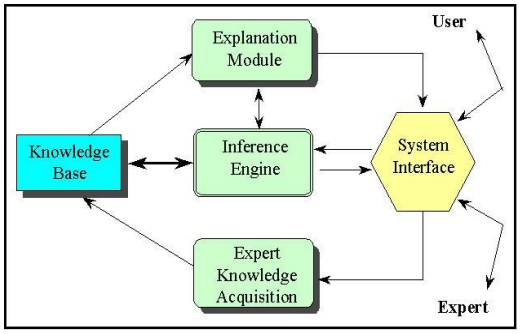

Fig.5. The Architecture

The knowledge base represents facts about the world. In early expert systems such as Mycin and Dendral these facts were represented primarily as flat assertions about variables. In later expert systems developed with commercial shells the knowledge base took on more structure and utilized concepts from object-oriented programming. The world was represented as classes, subclasses, and instances and assertions were replaced by values of object instances. The rules worked by querying and asserting values of the objects.
The inference engine is an automated reasoning system that evaluates the current state of the knowledge-base, applies relevant rules, and then asserts new knowledge into the knowledge base. The inference engine may also include capabilities for explanation, so that it can explain to a user the chain of reasoning used to arrive at a particular conclusion by tracing back over the firing of rules that resulted in the assertion (Kwak, 2012).

There are primarily two modes for an inference engine: forward chaining and backward chaining. The different approaches are dictated by whether the inference engine is being driven by the antecedent (left hand side) or the consequent (right hand side) of the rule. In forward chaining an antecedent fires and asserts the consequent. For example, consider the following rule:
A simple example of forward chaining would be to assert Man (Socrates) to the system and then trigger the inference engine. It would match R1 and assert Mortal (Socrates) into the knowledge base.

Backward chaining is a bit less straight forward. In backward chaining the system looks at possible conclusions and works backward to see if they might be true. So if the system was trying to determine if Mortal(Socrates) is true it would find R1 and query the knowledge base to see if Man(Socrates) is true. One of the early innovations of expert systems shells was to integrate inference engines with a user interface. This could be especially powerful with backward chaining. If the system needs to know a particular fact but doesn't it can simply generate an input screen and ask the user if the information is known. So in this example, it could use R1 to ask the user if Socrates was a Man and then use that new information accordingly.

The use of rules to explicitly represent knowledge also enabled explanation capabilities. In the simple example above if the system had used R1 to assert that Socrates was Mortal and a user wished to understand why Socrates was mortal they could query the system and the system would look back at the rules which fired to cause the assertion and present those rules to the user as an explanation. In English if the user asked "Why is Socrates Mortal?" the system would reply "Because all men are mortal and Socrates is a man". A significant area for research was the generation of explanations from the knowledge base in natural English rather than simply by showing the more formal but less intuitive rules.

As expert systems evolved many new techniques were incorporated into various types of inference engines. Some of the most important of these were:

- **Truth Maintenance** - Truth maintenance systems record the dependencies in a knowledge-base so that when facts are altered dependent knowledge can be altered accordingly. For example, if the system learns that Socrates is no longer known to be a man it will revoke the assertion that Socrates is mortal.

- **Hypothetical Reasoning** - In hypothetical reasoning, the knowledge base can be divided up into many possible views, a.k.a. worlds. This allows the inference engine to explore multiple possibilities in parallel. In this simple example, the system may want to explore the consequences of both assertions, what will be true if Socrates is a Man and what will be true if he is not?

- **Fuzzy Logic** - One of the first extensions of simply using rules to represent knowledge was also to associate a probability with each rule. So, not to assert that Socrates is mortal but to assert Socrates may be mortal with some probability value. Simple probabilities were extended in some systems with sophisticated mechanisms for uncertain reasoning and combination of probabilities.

- **Ontology Classification** - With the addition of object classes to the knowledge base a new type of reasoning was possible. Rather than reason simply about the values of the objects the system could also reason about the structure of the objects as well. In this simple example Man can represent an object class and R1 can be redefined as a rule that defines the class of all men. These types of special purpose inference engines are known as classifiers. Although they were not highly used in expert systems, classifiers are very powerful for unstructured volatile domains and are a key technology for the Internet and the emerging Semantic Web.

IV. APPLICATIONS, ADVANTAGES AND DISADVANTAGES OF EXPERT SYSTEMS

A. USES OF EXPERT SYSTEMS

1) Medical diagnosis (the knowledge base would contain medical information, the symptoms of the patient would be used as the query, and the advice would be a diagnose of the patient's illness). Such an expert-user interaction is illustrated in Fig. 6.

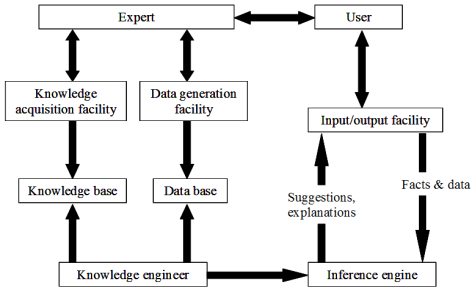

Fig.6. Expert-User Interaction

2) Playing strategy games like chess against a computer (the knowledge base would contain strategies and moves, the player's moves would be used as the query, and the output would be the computer's 'expert' moves)

3) Providing financial advice - whether to invest in a business, etc. (the knowledge base would contain data about the performance of financial markets and businesses in the past).

4) Helping to identify items such as plants / animals / rocks / etc. (the knowledge base would contain characteristics of every item, the details of an unknown item would be used as the query, and the advice would be a likely identification)

5) Helping to discover locations to drill for water / oil (the knowledge base would contain characteristics of likely rock formations where oil / water could be found, the details of a particular location would be used as the query, and the advice would be the likelihood of finding oil / water there)

6) Helping to diagnose car engine problems (like medical diagnosis, but for cars)

B. ADVANTAGES OF EXPERT SYSTEMS

The goal of knowledge-based systems is to make the critical information required for the system to work explicit rather than implicit. In a traditional computer program, the logic is embedded in code that can typically only be reviewed by an IT specialist. With an expert system, the goal was to specify the rules in a format that was intuitive and easily understood, reviewed, and even edited by domain experts rather than IT experts. The benefits of this explicit knowledge representation were rapid development and ease of maintenance.

Ease of maintenance is the most obvious benefit. This was achieved in two ways. First, by removing the need to write conventional code many of the normal problems that can be caused by even small changes to a system could be avoided with expert systems. Essentially, the logical flow of the program (at least at the highest level) was simply a given for the system, simply invoke the inference engine. This also was a reason for the second benefit: rapid prototyping. With an expert system shell it was possible to enter a few rules and have a prototype developed in days rather than the months or year typically associated with complex IT projects.

A claim for expert system shells that was often made was that they removed the need for trained programmers and that experts could develop systems themselves. In reality this was seldom if ever true. While the rules for an expert system were more comprehensible than typical computer code they still had a formal syntax where a misplaced comma or other character could cause havoc as with any other computer language. The expert system shell is shown in Fig.7.

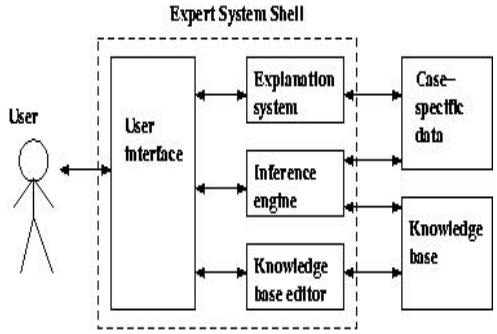

Fig.7. The Expert System Shell

In addition, as expert systems moved from prototypes in the lab to deployment in the business world, issues of integration and maintenance became far more critical. Inevitably demands to integrate with and take advantage of large legacy databases and systems arose. To accomplish this integration required the same skills as any other type of system.

C. DISADVANTAGES OF EXPERT SYSTEMS

The most common disadvantage cited for expert systems in academic literature is the knowledge acquisition problem. Obtaining the time of domain experts for any software application is always difficult but for expert systems it was especially difficult because the experts were by definition highly valued and in constant demand by the organization. As a result of this problem a great deal of research in the later years of expert systems was focused on tools for knowledge acquisition, to help automate the process of designing, debugging, and maintaining rules defined by experts. However, when looking at the life-cycle of expert systems in actual use other problems seem at least as critical as knowledge acquisition. These problems were essentially the same as those of any other large system: integration, access to large databases, and performance.

Performance was especially problematic because early expert systems were built using tools such as Lisp, which executed interpreted rather than compiled code. Interpreting provided an extremely powerful development environment but with the drawback that it was virtually impossible to match the efficiency of the fastest compiled languages of the time, such as C.

System and database integration were difficult for early expert systems because the tools were mostly in languages and platforms that were neither familiar to nor welcomed in most corporate IT environments –

programming languages such as Lisp and Prolog and hardware platforms such as Lisp Machines and personal computers. As a result, a great deal of effort in the later stages of expert system tool development was focused on integration with legacy environments such as COBOL, integration with large database systems, and porting to more standard platforms. These issues were resolved primarily by the client-server paradigm shift as PCs were gradually accepted in the IT world as a legitimate platform for serious business system development and as affordable minicomputer servers provided the processing power needed for AI applications.

V. CONCLUSION

In this paper, the architecture, structure and applications of Expert System, an AI sub-system was comprehensively reviewed and analysed. The core importance of Expert System in modern day businesses, industrial applications, medicine etc cannot be overemphasized; it is highly applied in different human disciplines nowadays and will still be useful for many years to come because it makes life easier for humans.

REFERENCES

Jackson Peter, “Introduction to Expert Systems”, (3rd Ed.), Addison Wesley Publishers, London, 2008, Pp. 2.

Leondes Cornelius T., “Expert Systems: the Technology of Knowledge Management and Decision Making for the 21st century”, IEEE Conference, 2009.

Russell Stuart, Norvig Peter, “Artificial Intelligence: A Modern Approach”, Simon & Schuster Publishers, Scotland, 2014, pp. 22–23.

Hayes-Roth Frederick, Waterman Donald, Lenat Douglas, “Building Expert Systems”, Addison-Wesley, London, 2011, Pp. 6–7.

Smith Reid, "Knowledge-Based Systems Concepts, Techniques and Examples", Schlumberger-Doll Research, Canada, 2013, Pp12-15.

MacGregor Robert, "Using a Description Classifier to Enhance Knowledge Representation", IEEE Expert, London, 2013, Pp. 4-7.

Wong Bok, Monaco John A., Monaco, "Expert System Applications in Business: A Review and Analysis of the Literature", Information and Management Journal, China, 2009, Pp. 141–152.

Kendal S.L., Green M., “An Introduction to Knowledge Engineering”, Springer, London, 2012, Pp.6-17.

Woolery L.K., Grzymala-Busse J., "Machine Learning for an Expert System to Predict Preterm Birth Risk", Journal of the American Medical Informatics Association, 2011, Pp.439–446.

Salvaneschi Paolo, Cadei Mauro, Lazzari Marco, "Applying AI to Structural Safety Monitoring and Evaluation", IEEE Expert - Intelligent Systems, London, 2009, Pp. 24–34.

Kwak S. H., "A Mission Planning Expert System for an Autonomous Underwater Vehicle", Proceedings of the 1990 Symposium on Autonomous Underwater Vehicle Technology, India, 2012, Pp.123–128.

www.jiengtech.com. All Rights Reserved.